

Express Mailing Label No. EL487542388US

PATENT APPLICATION

Docket No. 3000.2.28

IBM No. ST9-99-153

UNITED STATES PATENT APPLICATION

of

YUDONG SUN

for

SERVER-SIDE HTML CUSTOMIZATION BASED ON STYLE SHEETS AND A
TARGET DEVICE

MADSON & METCALF, P.C.

ATTORNEYS AT LAW
900 GATEWAY TOWER WEST
15 WEST SOUTH TEMPLE
SALT LAKE CITY, UTAH 84101

00112200 09321560

SERVER-SIDE HTML CUSTOMIZATION BASED ON STYLE SHEETS AND A TARGET DEVICE

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates generally to techniques for processing a hypertext markup language (HTML) document. More particularly, the present invention relates to a system and method for server-side HTML customization based on style sheets and a target device.

Identification of Copyright

A portion of the disclosure of this patent document contains material subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

Relevant Technology

The World Wide Web (hereinafter “the Web”) is a collection of Internet-accessible servers from which specially formatted documents may be retrieved and displayed by Web browsers, such as Netscape Navigator™ and Microsoft Internet Explorer™. Currently, the hypertext markup language (“HTML”) is the most

Unfortunately, providing target device-specific versions of Web pages usually means providing separate Web pages identified by different URLs, which is problematic for a number of reasons. For example, a Web page developer would need to create and maintain (e.g. update) several different Web pages, resulting in increased costs and the possibility of inconsistent versions. Moreover, separate indexes and links would need to be created for Web pages corresponding to various target devices, greatly increasing the sizes of current indexes and Web pages.

External data files containing style information are typically identified by a “.css” extension, e.g., “style.css.”

Unfortunately, many Web browsers do not support style sheet processing. For example, a PDA typically has a limited memory and central processing unit (CPU). Accordingly, PDA-based Web browsers are not able to process style sheets.

Accordingly, what is needed is a system and method for server-side HTML customization. What is also needed is a system and method for server-side HTML customization based on style sheets and a target device. Moreover, what is also needed is a system and method for maintaining one version of an HTML document for various types of target devices with different capabilities.

SUMMARY OF THE INVENTION

The present invention solves many or all of the foregoing problems by providing a system and method for server-side HTML customization based on style sheets and a target device.

In one aspect of the invention, a request reception module may receive a request for an document stored within document server. The document may be encoded in the hypertext markup language (HTML) and may include one or more HTML elements.

After the request is received, a parsing module may parse the requested document to generate therefrom a corresponding document object model (DOM) including at least one object. Each HTML element of the document typically corresponds to one objects of the DOM.

After the document is parsed, a style sheet access module may obtain a style sheet including at least one rule directed to a target device. In one embodiment, a target device identification module may identify the target device, and a style sheet identification module may identify at least one rule of a style sheet corresponding to the identified target device. In various embodiments, a single style sheet may contain rules for different target devices.

In another aspect of the invention, a style sheet application module may apply the identified style sheet rules to the DOM, after which a flattening module

These and other objects, features, and advantages of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is more fully disclosed in the following specification, with reference to the accompanying drawings, in which:

Figure 1 is a schematic block diagram of a computer system suitable for hosting a plurality of software modules according to an embodiment of the invention;

Figure 2 is a schematic block diagram of a system for server-side customization of a hypertext markup language (HTML) document based on style sheets and a target device according to an embodiment of the invention;

Figure 3 is schematic flowchart of a method for server-side HTML customization based on style sheets and a target device according to an embodiment of the invention;

Figure 4 is an illustration of an HTML document according to an embodiment of the invention;

Figure 5 is an illustration of a Document Object Model (DOM) according to an embodiment of the invention;

Figure 6 is an illustration of a style sheet according to an embodiment of the invention;

Figure 7 is an illustration of a transformed DOM according to an embodiment of the invention;

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 0
- 1

2
3
4
5
6
7
8
9

0
1
2

3
4
5
6
7
8

9
20
21

The computer workstation 12 may also include one or more input devices 26, such as a mouse and/or a keyboard, for receiving inputs from a user. Similarly, one or more output devices 28, such as a monitor and/or a printer, may be provided within, or be accessible from, the computer workstation 12.

In various embodiments, one or more Web servers 46 may be accessible to the workstations 12 via the Internet 40. A Web server 46 may be implemented using a workstation 12, as described above, including specialized software for delivering

1 (serving) Web pages to Web browsers. A variety of Web server application
2 programs are available, including public domain software from the National Center
3 for Supercomputing Applications (NCSA) and Apache, as well as commercial
4 packages from Microsoft, Netscape and others.

5 Referring now to Figure 2, a system 48 for server-side HTML customization
6 may include a Web server 46 and a target device 50. The target device 50 may be
7 implemented using a workstation 12, which includes a Web browser 52, such as
8 Netscape Navigator™ or Microsoft Internet Explorer™. The Web browser 52 may
9 be configured to communicate with the Web server 46 via the hypertext transfer
10 protocol ("HTTP").

11 In various embodiments, the target device 50 may include a standard
12 desktop computer, such as an IBM PC™ or compatible. In alternative embodiments,
13 however, the target device 50 may include a Web-enabled personal data assistant
14 (PDA), such as a PalmPilot™ VII, available from 3Com Corporation, or the like.

15 The Web server 46 is depicted as including a request reception module 54.
16 In one embodiment, the request reception module 54 receives (from the Web
17 browser 52) a request for a document 56 stored within a document storage area 58
18 of the Web server 46. The document 56 may be encoded in the hypertext markup
19 language ("HTML") and may include one or more HTML elements 57, as described
20 more fully hereafter.
21

1 In one embodiment, the Web server 46 also includes a parsing module 60,
2 commonly referred to as a "parser." The parsing module 60 retrieves, in various
3 embodiments, the requested document 56 and parses the document 56 to generate
4 therefrom a corresponding Document Object Model (DOM) 62, often referred to as
5 a "parse tree." A DOM 62 is a tree-like, hierarchical data structure including one or
6 more objects 64 that represent the various HTML elements 57 of the document 56.

7 In certain embodiments, the parsing module 60 is a conventional HTML
8 parser. For example, both Netscape Navigator™ and Microsoft Internet Explorer™
9 include HTML parsers, which may be adapted, in various embodiments, for use
10 within the Web server 46. In an alternative embodiment, a custom HTML parser
11 may be used. Conventionally, however, a Web server 46 does not include a parsing
12 module 60, since a document 56 is normally parsed only by a Web browser 52 at the
13 time the document 56 is displayed.

14 The Web server 46 may also include a style sheet access module 66. In
15 certain embodiments, the style sheet access module 66 is configured to retrieve a
16 style sheet 68 (from a style sheet storage area 70) including one or more rules
17 directed to a target device 50.

18 The style sheet access module 66 may include a target device identification
19 module 69, which may identify the type or class of the target device 50. This may
20 be accomplished, for example, based on platform information provided as part of
21

1 a browser request. Typically, a browser request includes a browser name and
2 version, as well as information about the platform, such as screen resolution.

3 The style sheet access module 66 may also include a style sheet identification
4 module 71. According to various embodiment, a single style sheet 68 may include
5 rules directed to different target devices 50. For example, rules directed to a PDA-
6 type device may be identified within the style sheet 68 by @media handheld
7 indicator or the like. Consequently, the style sheet identification module 71 may
8 identify the rules of the style sheet 68 corresponding to the identified target device
9 50.

10 The Web server 46 may also include a style sheet application module 72,
11 which applies the appropriate rules of the style sheet 68 to the DOM 62 of the
12 document 56. Techniques for applying style sheets rules are well known in the art.
13 For example, both Netscape Navigator™ and Microsoft Internet Explorer™ include
14 style sheet application modules 72, which may be adapted, in various embodiments,
15 for use within the Web server 46. In an alternative embodiment, however, a custom
16 style sheet application module 72 may be used.

17 In one embodiment, the style sheet access module 66 includes an object
18 removal module 74. Where, for instance, a rule within a style sheet 68 indicates a
19 "NONE" display style, or similar designation, for an element 57 or element type, a
20 corresponding object 64 within the DOM 62 is preferably removed.
21

1 For example, the rule, `IMG { display: NONE }`, indicates a "NONE"
2 display style for the IMG (in-line image) element type. Accordingly, the object
3 removal module 74 preferably removes the object(s) 64 of the DOM 62
4 corresponding to in-line image elements 57. This is advantageous, for instance,
5 where a document 56 includes in-line images, but a target device 50, such as a PDA,
6 cannot display such images.

7 The style sheets 68 and the Web documents 56 are depicted as logically
8 separate data files, and may even be stored within separate storage areas 58, 70 of
9 the Web server 46. In an alternative embodiment, a style sheet 68 may be included
10 within a separate portion of the document 56. For example, the HTML elements 57
11 of the document 56 and the rules of the style sheet 68 may be stored within separate
12 portions of a single logical data file.

13 The Web server 46 may also include a flattening module 76. In various
14 embodiments, the flattening module 76 flattens the DOM 62 to generate therefrom
15 a corresponding transformed document 78. As used herein, the term "flattening"
16 refers to a process of converting the DOM 62 back into an equivalent HTML
17 document 86 including one or more corresponding HTML elements 57. Techniques
18 for flattening a DOM 62 are well known in the art. The resulting document 86
19 is designated as "transformed" because the style sheet application will be reflected
20 in the HTML elements 57 of the transformed document 78.

21

1 In various embodiments, the Web server 46 may also include a transmission
2 module 80. The transmission module 80 may send the transformed document 78
3 (via the Internet 40) to the Workstation 12, such that the document 86 may be
4 displayed by the Web browser 52.

5 Referring now to Figure 3, a schematic flowchart includes a method 100 for
6 server-side HTML customization according to a presently preferred embodiment
7 of the invention. The method 100 may begin by receiving 102, at a Web server 46,
8 a request for a document 56.

9 Figure 4 illustrates an exemplary document 56 according to an embodiment
10 of the invention. The document 56 may include one or more HTML elements 57,
11 such as a paragraph element 57A and an image element 57B.

12 After the document request is received 102, the method 100 may continue
13 by parsing 104 the document 56 to generate therefrom a corresponding Document
14 Object Model (DOM) 62. As noted, a DOM 62 is a tree-like, hierarchical data
15 structure including one or more objects 64 that represent the HTML elements 57 of
16 the document 56. Figure 5 illustrates a portion of a simplified DOM 62
17 corresponding to the document 56 of Figure 4.

18 After the document 56 is parsed 104, the method 100 may continue by
19 identifying 106 a target device 50 for displaying the document 56. As noted, the
20 target device 50 may be based on platform information provided by a browser
21 request.

Figure 6 illustrates an exemplary style sheet 68 for a PDA-type target device 50 according to an embodiment of the invention. The style sheet 68 may include any number of standard rules 72, such as rule-sets and at-rules (as defined in the CSS standard).

For example, a first rule 73A, i.e. `P { font-size: 10pt }` may set the font size for each paragraph element 57. Specifically, the rule 73A may set the font size to 10 points.

1 A second rule 73B, i.e. `IMG { display: NONE }`, may not include a
2 typical style declaration, but may specify "NONE" display style or a similar
3 designation. In various embodiments, a "NONE" display style causes the object
4 removal module 74 to remove objects 64 corresponding to the element type
5 specified in the rule 73.

6 After the style sheet 68 is identified, the method 100 may continue by
7 applying 110 the identified style sheet rules 73 to the DOM 62. Each rule 73 of the
8 style sheet 68 may be applied to the objects 64 of the DOM 62, which may result in
9 the removal of certain objects 64 and the addition of others.

10 For example, as illustrated in Figure 7, the rule 73A may add a new object
11 64E, corresponding to a `` element 57. By contrast, the rule 73B
12 may cause objects 64A-C (IMG elements 57) of Figure 5 to be deleted. After
13 application of the style sheet 68, the DOM 62 may appear as shown in Figure 7.

14 While the style sheet 68 and the document 56 are depicted herein as logically
15 separate data files, the style sheet 68 may be included, in some instances, within a
16 separate portion of document 56. For example, all of the rules 73 of the style sheet
17 68 may be located, as a group, at the beginning of the document 56:

```
18 <style>  
19     P { font-size: 10pt }  
20     IMG { display: NONE }  
21 </style>  
<html>  
<head>  
<TITLE>A Simple HTML Document</TITLE>
```

• • •

```
@media handheld
{
P { font-size: 10pt }
IMG { display: NONE }
}

@media tinyscreen
{
P { font-size: 12pt }
IMG { display: NONE }
}
```

In such an embodiment, the style sheet access module 66 may parse the style sheet 68 and extract the rules 73 corresponding to the identified target device 50.

After the rules 73 have been applied 110, the method 100 may continue by flattening 112 the DOM 62 to create a transformed document 78, which may then be sent 114 to the requesting Web browser 52 for display. As previously noted, the flattening process involves converting the DOM 62 back into an HTML document 86. Consequently, any transformations to the DOM objects 64 will be preferably reflected in the corresponding HTML elements 57 of the document 86.

For example, Figure 8 illustrates an exemplary transformed document 78 after flattening 116 the DOM 62 of Figure 7. Comparing the transformed document

By contrast, the system and method of the present invention apply style sheets 68 on the Web server 46. Server-side HTML customization results in a more compact document 56 that may be sent to a target device 50 over a limited-

1 bandwidth network. Moreover, the need for bandwidth is further reduced because
2 the style sheets 68 are never sent to the target device 50.

3 The present invention may be embodied in other specific forms without
4 departing from its scope or essential characteristics. The described embodiments
5 are to be considered in all respects only as illustrative and not restrictive. The scope
6 of the invention is, therefore, indicated by the appended claims rather than by the
7 foregoing description. All changes which come within the meaning and range of
8 equivalency of the claims are to be embraced within their scope.

9 What is claimed is:
10
11
12
13
14
15
16
17
18
19
20
21